SGS-THOMSON MICROELECTRONICS

ACTUM REALIZER® FOR THE ST62

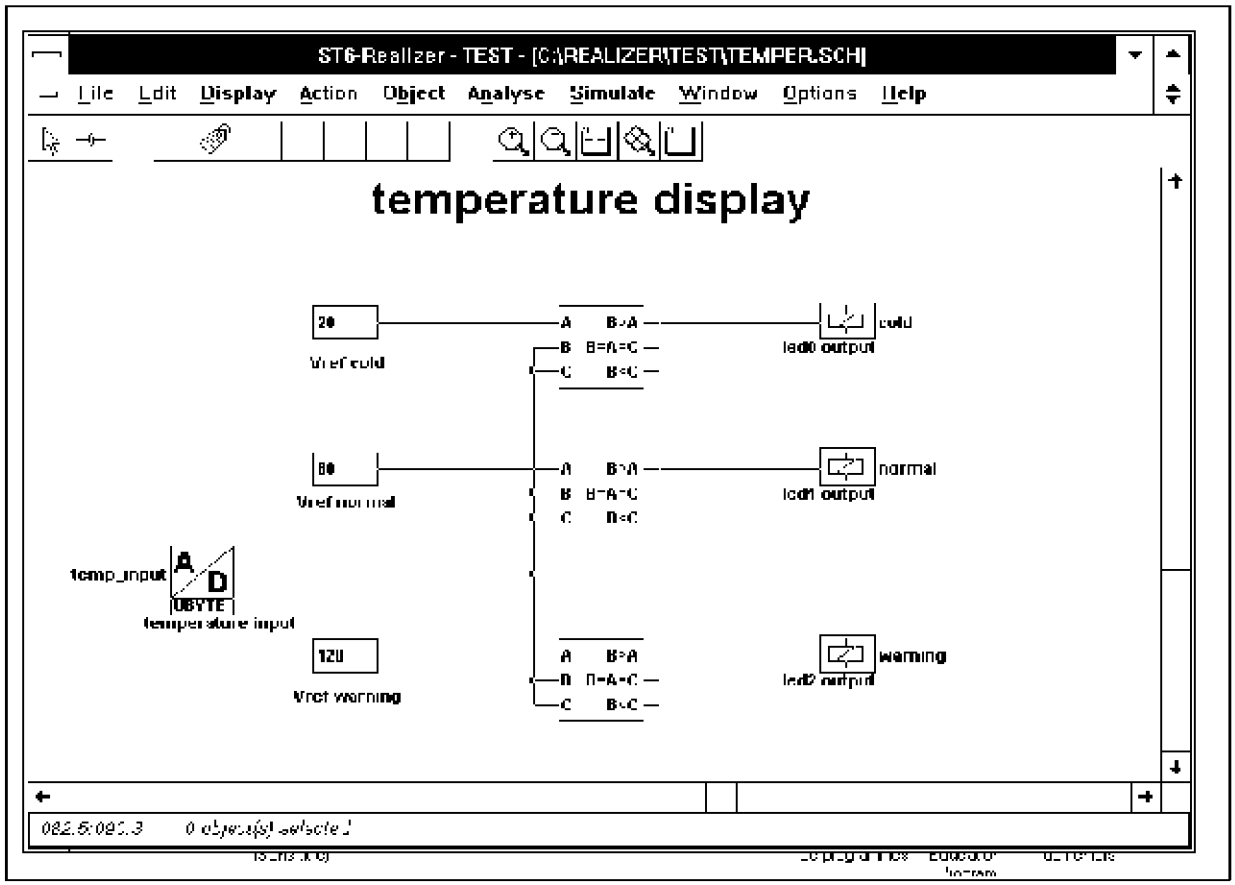
SOFTWARE AIDED COMPUTER ENGINEERING FOR ST62 8-BIT MICROCONTROLLER

PRELIMINARY DATA

GRAPHIC DESIGN AND DEBUG

- SCHEMATIC-BASED SOFTWARE DESIGN
  - Industry Standard Graphic symbols
  - Extensive Symbol Library
  - Select and Wire on-screen to generate Application
  - Built-In Self-Documentation
- SCHEMATIC-BASED ANALYSIS
  - Single click Analyze Operation
  - Path and Functional Verification
  - Efficient code generation for ST62
- SCHEMATIC-BASED SIMULATOR FOR DEBUG
  - Runs on design schematic
  - Stimulate and Observe On-line
  - Add Virtual tools (Oscilloscope, Pulse, Time, Data Generators, logic probes)
  - Uses ST62 Code from Analyzer
  - Dedicated to ST62 Microcontroller

Figure 1. Schematic Entry of a simple Application



DESCRIPTION

The ST62-Realizer is a version of The Realizer® by Actum Solutions [1], The Netherlands, dedicated to design and development of applications for the ST62 family of microcontrollers (MCUs).

The schematic-based architecture for both design and Simulation allows all engineers, even without Microcontroller experience, to create a Microcontroller application, with all the benefits and flexibility that microcontrollers offer.

DESIGN

Use your knowledge of the application to decide Input/Output functions, then draw and build up the application functionality graphically using the industry standard symbols from the library (or create your own). A state machine can be added if required. Draw wires between the symbols on-screen to create the application. The schematic can be printed for documentation.

ANALYSE

Select an ST62 device and allocate I/O functions to actual “physical” I/O pins. Run the Analyzer to verify and run process analysis on the application as drawn. If errors are found, edit the schematic and re-analyze. When all is satisfactory, allow the Analyzer to generate the efficient ST62 software code.

DEBUG

Use the integrated ST62 Symbolic Schematic Simulator to stimulate and monitor the circuit functionality directly on the schematic. Add Virtual development tools to the circuit as graphical symbols and run or step through the functionality to verify that the application works as expected. If not, return to the Schematic entry environment and revise the drawing, then Analyse and Debug again until the function is as desired.

Now you can easily program EPROM or OTP ST62 microcontrollers with the final generated software to build prototypes. Field trials can then be performed with confidence that the application program functionality has been assured, giving less time needed for expensive field modifications.

EXPANSION

The Realizer can produce a complete standalone application program, or just the main part to startup and control the program functions designed on-screen. In this latter case, the code can be linked with other library routines (for example Fuzzy Logic routines) to expand on the functionality. The expanded code is then able to be debugged with the ST62 Hardware Emulation Tools.

HARDWARE/SOFTWARE REQUIREMENTS

- A 80386 (or higher) PC with at least 2Mbytes of memory
- MS-Windows 3.0 or higher and MS-DOS 3.3 or higher
- Hard disk with 5MB of free disk space and a 3.5” floppy disk drive
- VGA monitor supported by Windows and a pointing device (e.g. a mouse)

Note [1]: Actum Solutions:
PO Box 373
1700 HJ HEERHUGOWAARD
The Netherlands

Figure 2. Adding functional Blocks as Symbols

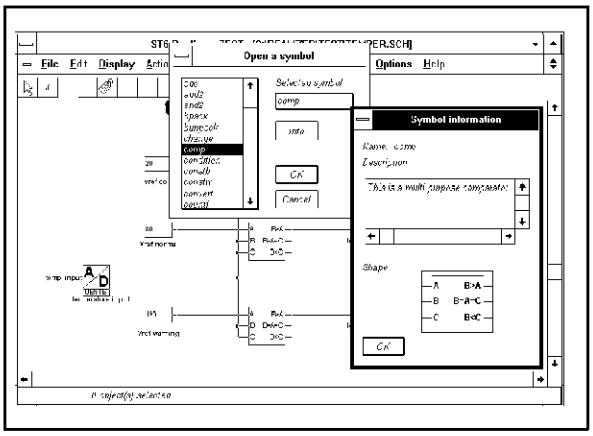
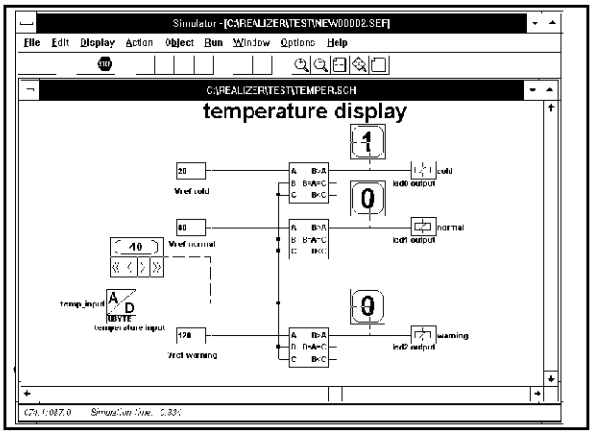


Figure 3. Testing with virtual Tools



Extract of available Symbols and Functions

adc	Analog to digital converter
add2	Two input adder with type inheritance
and2	Two input bitwise AND function, with type inheritance
bpack	Eight bits to one byte packer
bunpack	One byte to eight bits unpacker
change	Change detector.
comp	Multi purpose comparator
condition	Condition function for the state machines
constb	Constant bit symbol
constw	Constant word symbol
convert	Type conversion symbol
countf	Counter with a fixed preset value
countv	Counter with a variable preset value
dac	Digital to analog convertor
delf	Delay with a fixed on and off delay time
delfoff	Delay with a fixed off delay time
delfon	Delay with a fixed on delay time
delv	Delay with a variable on and off delay time
delvoff	Delay with a variable off delay time
delvon	Delay with a variable on delay time
dff	D-flipflop with a multiple type data input
digin	Digital input
digout	Digital output
div	2 input divider with quotient and remainder output, with multi type inheritance
edge	Rising edge detector
indextable	Index table, the input is used as the index in a table
init	Output only TRUE during the first loop after a reset
inv	Multi type bitwise inverter
limf	Fixed limiter, the output will not be larger than the top value and not smaller than the bottom value
limv	Variable limiter, the output will not be larger than the top value and not smaller than the bottom value
lookuptable	Look up table, input value is used to search through a table to find the output value
loopdel	Output always the previous value of the current input (delays one loop)
mul	2 input multipier, multi type inheritance
mux1	Two input multiplexer. If the selection input is FALSE input 0 is copied to the output, otherwise input 1 is copied
mux2	Four input multiplexer
or2	Two input OR function, multi type inheritance
oscf	Fixed time oscillator, frequency equals: f(Hz) = 1/(2*time)
oscv	Variable time oscillator, frequency equals: f(hz) = 1/(2*time)
portin	Connect sub scheme symbol pins from the parent scheme with the sub scheme nets
portout	Connect sub scheme symbol pins from the parent scheme to the sub scheme nets
shift	Shift register symbol with parallel in, serial in, shift up, shift down
srff	Set/reset flip flop
sss	Example of a sub scheme symbol
state	State symbol, used within a state machine
statein	State input symbol to connect to a state machine
stateinit	Initial state of a state machine
stateout	State output symbol for extracting a state from a state machine
sub2	Two input subtractor, with type inheritance
timf	Fixed timer, which will generate a pulse on a rising edge on the input
timv	Variable timer, which will generate a pulse on a rising edge on the input
title	Title block is used for archiving purposes
wmerge	Build a word out of a high byte and a low byte
wpack	Pack sixteen bits into one word
wsplit	Split a word into a high byte and a low byte
wunpack	Unpack a word into sixteen bits
xor	Two input bitwise exor function, multi type inheritance

ORDERING INFORMATION

Sales Type	Description
ST6-REALIZER/PC	Software Aided Computer Engineering for ST62 8-bit Microcontroller, Microsoft Windows 3™ Edition